IN THE CLAIMS

Listing of the Claims:

1. (Currently Amended)   A naming service for locating a service in an enterprise, comprising:

      a first module operable to maintain a location of an interface, the interface having a

           reference to a service; and

      a second module operable to provide the location of the interface to an application in

           response to receiving a request from the application for the location of the service;

           <u>wherein the naming service provides service location transparency such that the location</u>

           <u>of the service can be changed without effecting the behavior of the application</u>.

2. (Original)   The naming service of Claim 1, wherein application is operable, using the location of the interface, to the service using the interface.

3. (Original)   The naming service of Claim 1, wherein the service is further defined as a service object.

4. (Currently Amended)   The naming service of Claim 3, wherein the service object is further defined as a [[Java]] <u>JAVA</u> service object and wherein the interface is further defined as a [[java]] <u>JAVA</u> directory and naming interface.

2

5. (Currently Amended)   The naming service of Claim 3, wherein the service object is selected from a [[goup]] group of service objects including an enterprise [[Java]]JAVA Bean, a queue, and a queue manager.

6. (Original)   The naming service of Claim 1, wherein the first module is further operable to maintain a second location associated with a second service, the second module further operable to provide the second location to a second application in response to receiving a request for the second service from the second application, the second application using the second location to use the service.

7. (Original)   The naming service of Claim 6, wherein the second service is a service object and wherein the second location is useful by the application to directly invoke the services.

8. (Original)   The naming service of Claim 7, wherein the second location is selected from a group of locations including an address and reference location.

9. (Original)   The naming service of Claim 1, wherein first module further maintains an identifier corresponding to the service and associates the identifier with the location of the interface.

10. (Original) The naming service of Claim 9, wherein the identifier is a service type of the service.

11. (Original) The naming service of Claim 9, wherein the identifier includes is a name and a service type associated with the service.

12. (Original) The naming service of Claim 1, wherein the interface is a server and the application is a client of the server, the client using the server to provide the service to the client.

13. (Original) The naming service of Claim 1, wherein the second module returns meta-information to the application, the meta-information including a reference useful by the application for employing the service.

14. (Original) The naming service of Claim 1, wherein the service is a server maintaining a plurality of classes and a plurality of objects, at least one of the objects useful by the application.

15. (Original) The naming service of Claim 1, wherein first module stores the location of the interface in a datastore.

16. (Original) The naming service Claim 15, wherein the datastore is further defined as a lightweight directory access protocol based datastore.

17. (Original) The naming service of Claim 1, further including a third module operable to store a service status information related to the service, the third module operable to search and return the service status information related to the service in response to a request.

4

18. (Original) The naming service of Claim 17, wherein a hypertext markup language interface is employed to communicate with the third module.

19. (Original) The naming service of Claim 18, wherein the third module is defined as a name service browser.

20. (Currently Amended)   An enterprise naming service for applications to locate services, comprising:

> a binding module to associate a first service with a location of an interface maintaining a reference to the first service, the binding module further operable to associate a second service with a location <u>of an interface maintaining a reference to</u> [[of]]the second service; and

> a look-up module [[operative]] <u>operable</u> to provide the location of the interface <u>of the first service to a first application</u> in response to a request by [[an]] <u>the first</u> application for the first service, the look-up module further operable to provide the location <u>of the interface</u> of the second service in response to a request by a second application <u>for the second service;</u>

> <u>wherein the enterprise naming service provides service location transparency such that the location of a service can be changed without effecting the behavior of an application.</u>

21. (Original) The enterprise naming service of Claim 20, wherein the first service is a service object and the second service is a Common Object Request Broker Architecture object.

22. (Currently Amended)   The enterprise naming service of Claim 21, wherein the interface is a [[Java]] <u>JAVA</u> Naming and Directory Interface and wherein the first service is a [[Java]] <u>JAVA</u> service object.

23. (Original) The enterprise naming service of Claim 21, further comprising a name service browser module operable to maintain a service status information related to one of the first and second services, the name service browser operable to search and return the service status information of one of the first and second services in response to a request.

24. (Original) The enterprise naming service of Claim 21, wherein the binding module is further operable to maintain a version identifier associated with at least one of the first and second services, and wherein a plurality of versions of at least one of the first and second services are maintained.

25. (Original) The enterprise naming service of Claim 24, wherein the look-up module is further operable to return the location associated to a first version of the first service.

26. (Original) The enterprise naming service of Claim 25, wherein the look-up module is further operable to return the location associated to a first version of the second service.

27. (Currently Amended)   A method for locating a service in an enterprise, comprising:

associating a service with a location with an interface maintaining a reference to a service;

requesting, by an application desiring to employ the service, the location of the service; and

returning the location of the interface to the application;

wherein the method provides service location transparency such that the location of the service can be changed without effecting the behavior of the application.

28. (Original) The method of Claim 27, further comprising:

using the location to communication between the application and the interface;

requesting, by the application, the service from the interface; and

using the service by the application.

29. (Original) The method of Claim 27, wherein the application uses a service identifier to request the location of the service.

30. (Original) The method of Claim 29, wherein the service is defined as a service object and wherein the interface is further defined as a naming and directory interface.

8

31. (Currently Amended)   The method of Claim 30, wherein the interface is defined as a [[Java]] JAVA Naming Directory Interface and the service is an Enterprise [[Java]] JAVA Bean, the method further comprising associating an identifier, a version and a second location with a Common Object Request Broker Architecture object.

32. (Original) The method of Claim 31, wherein the method further comprises:

>   requesting the second location of the Common Object Request Broker Architecture object using the identifier and version of the Common Object Request Broker Architecture object;

>   returning the second location of the Common Object Request Broker Architecture object based on the identifier and version of the Common Object Request Broker Architecture object;

>   connecting to the Common Object Request Broker Architecture object using the second location; and

>   employing the Common Object Request Broker Architecture object at the second location.

9